# Run tutorials

In docker:

```
cd PorePy
git pull
pip install –e .
```

Download tutorials:

```
cd /host
mkdir porepy; cd porepy
git clone https://github.com/keileg/porepy_intro
```

Use tutorials. Either:

```
jupyter nbconvert –to script name_of_tutorial.ipynb   # will give code, with a lot of comments surrounding it
ipython name_of_tutorial.py
```

Or:

Cut and paste python code into a python script

Or (if jupyter is already installed on host):

jupyter notebook (On the host, not docker)

# Exercise 1

| Frac_num | Start | End | Permeability |
|---|---|---|---|
| 0 | (0.2, 0.3) | (0.7, 0.3) | 1e-4 |
| 1 | (0.5, 0.1) | (0.8, 0.5) | 1e4 |
| 2 | (0.1, 0.4) | (0.5, 0.8) | 1e4 |

Solve a mixed-dimensional elliptic problem defined on the unit square, with fracture coordinates and permeabilities as defined in the table.

Create a grid with about 500 cells.

Matrix permeability: 1.

Boundary conditions as defined in MixedDimensionalFlow.ipynb.

For the normal diffusivity (on interfaces / edges) between 2d and 1d domains, use the same values as the fracture permeability.

In the intersection between fractures 0 and 1, try both normal diffusivity of 1e4 and 1e-4.

Hint: Identify individual fractures by g.frac_num

# Exercise 2

Reuse the setup from Ex. 1 (with normal diffusivity in fracture intersection set to 1e4). Introduce the following modifications:

- Assign homogeneous Neumann conditions on the entire global boundary.
- Assign a source term (pp.ScalarSource) of weight 1 in (0.3, 0.3) (the intersection), and a source with weight -1 in (0.3, 0.6) (fracture 2)

# Exercise 3

Reuse the setup from ex. 1, with the following modifications:

- Use an RT0 (pp.RT0) in fractures 0 and 2. What happens to the number of dofs in the subsystems for these subdomains (hint: see final cell of the notebook)
- Reintroduce the sources from ex. 2. For the source in the fracture, you will need to introduce a pp.DualScalarSource, to fit with the number of dofs in the subdomain.

4. Flow problem on 3d geometry
5. Flow problem with non-matching grids
6. Transport problem with source inside fracture.

6. K -> k / mu(T), mu = exp(-T). Rediskretiser (tpfa). Explisit koblet.

# Exercise 4 - setup

Solve a problem geometry on the unit cube. The vertexes of the fractures are defined in the table.

Matrix permeability: 1

Fracture permeability (also intersections and normal diffusivities): 1e4

Boundary conditions: u = 1 on x=0, u=0 on x=1, homogeneous Neumann otherwise.

| frac_num | coord_0 | Coord_1 | Coord_2 | Coord_3 | Coord_4 |
|---|---|---|---|---|---|
| 0 | (0.5, 0.0, 0.0) | (0.5, 0.8, 0.0) | (0.5, 0.7, 0.5) | (0.5, 0.2, 0.6) | |
| 1 | (0.2, 0.2, 0.5) | (0.8, 0.2, 0.5) | (0.8, 0.6, 0.5) | (0.5, 0.8, 0.5) | (0.2, 0.6, 0.5) |
| 2 | (0.3, 0.3, 0.3) | (0.7, 0.7, 0.3) | (0.7, 0.7, 0.7) | (0.3, 0.3, 0.7) | |

# Exercise 4 – tasks

1. Create a grid for the geometry, aim for roughly 1k cells. Test sensitivity of the cell count to the mesh size parameters.
2. Discretize a flow problem, using Mpfa discretizations.
3. Create a finer grid of fractures and intersections (hint: speedup mesh construction with network.mesh(dfn=True)). Create a non-matching grid by replacing grids in the original grid_bucket. Check the impact on the total cell count. (hint: a new pp.Assembler is needed for the new GridBucket).

# Exercise 5:

Consider the coupled flow-transport problem, on the form

$$-\nabla \cdot \left( \frac{K}{\mu(T)} \nabla p \right) = 0$$

$$q = -\frac{K}{\mu(T)} \nabla p$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (qT) - (D\nabla T) = 0$$

$$u = \exp(T)$$

# Ex. 5

Use the geometry defined in exercise 1.

Take the boundary conditions for flow as in ex. 1. For temperature, let T=1 on x=0, T=0 on x=1, homogeneous Neumann conditions on the rest.

Define a discretization scheme for the coupled system by extending the advection-diffusion problem.

Use an explicit coupling between flow and transport (solve flow equation with temperature from previous time step, then transport with the new velocity field).

Use pp.Tpfa for the flow problem. Note that the flow equation must be updated for every time step due to the viscosity term.